8Б

УДК 681.322

О.Ю. Бандура, А.Л. Головинский, С.Г. Рябчун
Институт кибернетики им. В.М. Глушкова НАН Украины, г. Киев, Украина
icybcluster@gmail.com

Построение высокопроизводительной дисковой подсистемы для суперкомпьютеров кластерной архитектуры

Исследованы общие концепции высокопроизводительной дисковой подсистемы как непременной составной части суперкомпьютеров кластерной архитектуры, а также рассмотрено ее построение в составе кластерного комплекса СКИТ.

Введение

Современные высокопроизводительные суперкомпьютеры кластерной архитектуры кроме надежного централизованного хранилища данных на основе параллельной файловой системы для своей эффективной работы также требуют быстродействующую временную дисковую систему на основе выделенных дисковых серверов, которая нужна для следующих целей:

1. Во время своей работы узлы вычислительного кластера требуют очень много виртуальной памяти. Обычно она размещается в разделе подкачки жесткого диска (*swap partition*).

2. Большинство кластерных вычислительных задач в процессе своей работы производят промежуточные результаты, которые сохраняются во временных файлах на диске. И быстродействие таких программ значительно увеличилось, если бы эти файлы сохранялись во временной файловой системе с высоким быстродействием.

Конечно, для таких целей можно использовать отдельный локальный жесткий диск для каждого узла кластера, но такое решение имеет ряд недостатков:

1. Отдельный жесткий диск, сколь бы производительным он ни был, не в состоянии обеспечить необходимое пиковое быстродействие для временной файловой системы.

2. Неэффективное использование оборудования, поскольку если сгруппировать жесткие диски в одном или нескольких дисковых серверах, то их можно использовать для нужд кластера в наиболее полном объеме, причем для достижения достаточного пикового быстродействия необходимо значительно меньшее количество самих жестких дисков на кластер.

3. Экономическая нецелесообразность – такой подход приводит к значительному удорожанию кластерного комплекса, при этом без реализации требуемого быстродействия.

4. Уменьшается надежность кластера, поскольку огромное число несгруппированных локальных жестких дисков является дополнительной точкой отказа, а в дисковых серверах можно использовать дополнительные методы отказоустойчивости, например дисковые *RAID массивы*.

Рассмотрим детальнее основную концепцию дисковой подсистемы кластера.

Концепция дисковой подсистемы

Основная концепция дисковой подсистемы кластера следующая. Существует отдельный дисковый сервер, который имеет массив жестких дисков. Он по сети экспортирует так называемые виртуальные диски, для каждого узла кластера – свой отдельный. Каждый узел подсоединяет по сети свой виртуальный жесткий диск, и работает с ним так, словно бы он являлся его собственным физическим диском. Все операции чтения / записи при этом транслируются по сетевой среде на дисковый сервер. Такой подход позволяет использовать бездисковые вычислительные узлы кластера, к тому же их быстродействие будет не хуже, а во многих случаях даже лучше, чем у аналогичных узлов с локальными жесткими дисками.

При этом для дисковой подсистемы кластера более важным будет максимальное быстродействие при работе с одним клиентом, чем ее общая производительность при одновременном обслуживании многих узлов, что более критично для централизованного хранилища данных кластера, поскольку способ использования виртуальной памяти и временной файловой системы не предусматривает массовые операции чтения/записи.

Возможно, наиболее известной технологией для построения дисковых систем является протокол *Internet SCSI (iSCSI)* [1]. Применение этого сквозного (end-to-end) протокола позволяет транспортировать блоки данных *SCSI* устройств между клиентом и сервером, используя стек *TCP/IP* (рис. 1). Преимуществом данной технологии является простота и дешевизна реализации, поскольку при этом задействуется существующая инфраструктура сети *Ethernet*. Но последнее порождает и главные недостатки, а именно, недостаточное быстродействие (максимальная полоса пропускания сети Ethernet составляет 1 Гбит/с с высокой латентностью передачи данных), а также низкая отказоустойчивость. Поэтому она не получила широкого применения в сфере кластерных суперкомпьютеров.

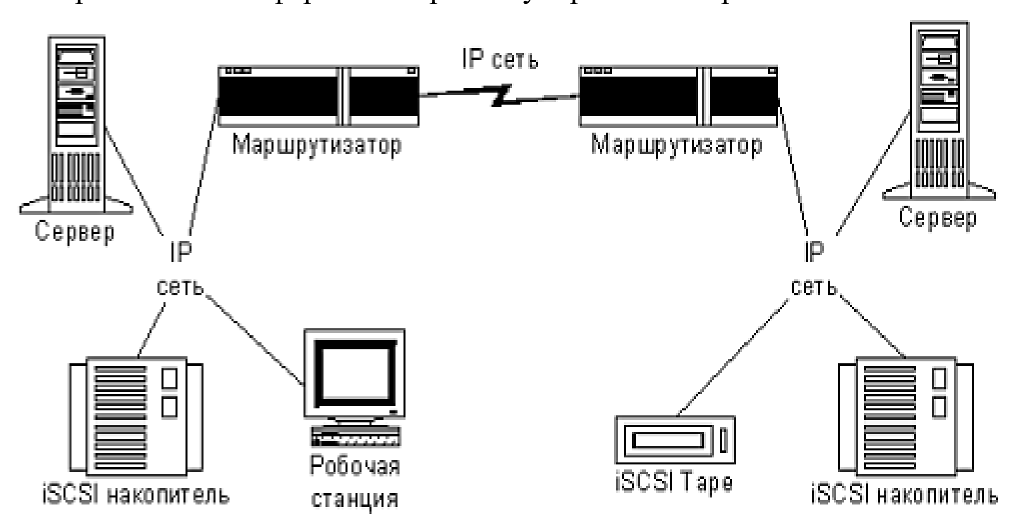


Рисунок 1 – IP сеть с использованием iSCSI устройств

Отдельно следует выделить такую технологию как файловая система *tmpfs* [2] под операционной системой Linux. Эта файловая система позволяет сохранять временные данные в виртуальной памяти компьютера. Подобно файловой *ramdisk*, tmpfs использует оперативную память, но, кроме этого, может использовать *swap devices* – устройства подкачки. В то время как традиционный ramdisk – это блочное

устройство и перед его использованием необходимо отформатировать раздел командой *mkfs* с опциями, файловая система tmpfs – устройство не блочное и готово к использованию сразу после монтирования. Такие свойства tmpfs делают ее наиболее привлекательной из *RAM-based* файловых систем, известных на сегодняшний день.

Ядро Linux «понимает» ресурс «виртуальная память» именно как единое целое – *RAM и swap devices*. Подсистема виртуальной памяти ядра предоставляет эти ресурсы другим подсистемам. При этом часто без ведома «подсистемы – заказчика» перемещает страницы оперативной памяти в swap, а при необходимости возвращает их обратно. Файловая система tmpfs использует страницы подсистемы виртуальной памяти для сохранения файлов. При этом сама tmpfs не знает, находятся ли эти страницы в swap или в RAM, это задача ядра Linux.

Все эти характеристики делают tmpfs идеальным вариантом для быстродействующей файловой системы для временных файлов промежуточных вычислений на узлах кластера. При этом дисковая подсистема кластера обеспечивает узлы скоростной виртуальной памятью благодаря разделу подкачки на примонтированных виртуальных дисках, поскольку самого объема оперативной памяти для таких целей будет недостаточно.

Обзор протокола SRP

В данной статье рассматривается как основа для построения дисковой подсистемы кластера новейшая технология *SCSI RDMA Protocol – SRP* [3]. Она очень похожа на протокол iSCSI, рассмотренный выше, но основным отличием является то, что *SCSI-блоки* данных транспортируются посредством коммуникационной сети с прямым удаленным доступом к памяти. Этот протокол детально описан в стандарте ANSI INCITS 365 – 2002.

RDMA (Remote Direct Memory Access) [4] – это группа протоколов удалённого прямого доступа к памяти, при котором передача данных из памяти одного компьютера в память другого компьютера происходит без участия операционной системы. При этом исключается участие центрального процессора в обработке кода переноса и необходимость пересылки данных из памяти приложения в буферную область операционной системы, то есть данные пересылаются напрямую на соответствующий сетевой контроллер. RDMA поддерживает множество сетевых интерконнектов, таких, как Myrinet, Infiniband, Quadrics и др. На рис. 2 изображена модель потока данных RDMA посредством коммутационной сети Infiniband.

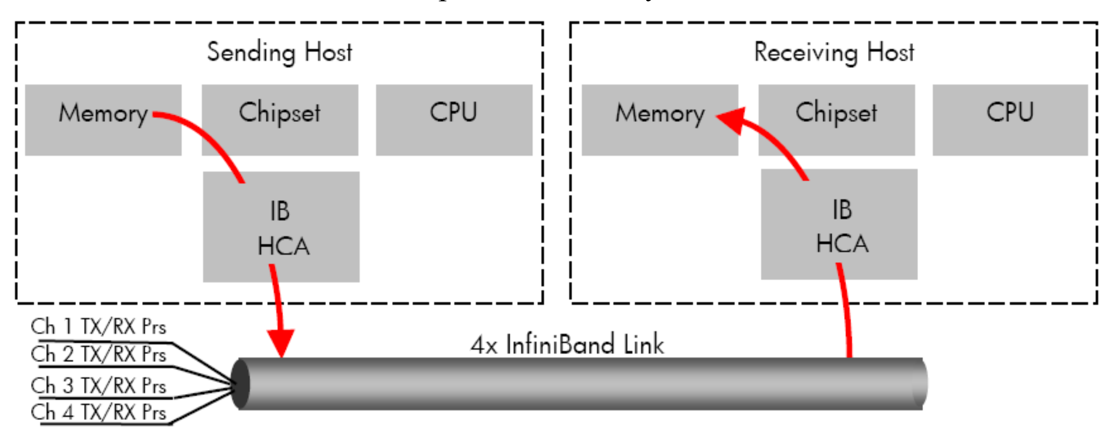


Рисунок 2 – RDMA поток данных через Infiniband

Архитектура обычного SCSI базируется на клиент-серверной модели. «Клиент», которым может быть, например, физический сервер, или рабочая станция, инициирует запросы на считывание или запись данных с исполнителя – «сервера», в роли которого, как правило, выступает система хранения данных. Команды, которые выдает «клиент» и обрабатывает «сервер», помещаются в блок описания команды (*Command Descriptor Block, CDB*).

CDB – это структура, с помощью которой приложение-клиент направляет команды устройству-серверу. «Сервер» выполняет команду, а окончание ее выполнения обозначается специальным сигналом. Инкапсуляция и надежная доставка CDB-транзакций между инициаторами и исполнителями через RDMA канал и есть главная задача SRP, причем ее приходится осуществлять в нетрадиционной для SCSI среде.

Протокол SRP осуществляет контроль передачи блоков данных и обеспечивает подтверждение достоверности завершения операции ввода-вывода, что в свою очередь обеспечивается через одно или несколько RDMA-соединений.

Чаще всего в качестве коммуникационной среды RDMA используется высокоскоростная сеть *Infiniband* [5]. Это архитектура коммутации соединений типа «точка-точка». Каждая линия связи представляет собой четырехпроводное двунаправленное соединение с пропускной способностью 2,5 Гбит/с в каждом направлении и гибким выбором физических линий передачи. В архитектуре InfiniBand определен многоуровневый протокол (физический, канальный, сетевой и транспортный уровни) для реализации аппаратными средствами, а также программный уровень для поддержки управляющих функций и скоростного обмена данными (с малыми задержками) между устройствами. Среди главных достоинств архитектуры InfiniBand – ее способность обеспечить за пределами сервера такую же производительность передачи данных, как и внутри него. Перечислим основные характеристики технологии InfiniBand:

а) возможность масштабирования пропускной способности линий связи до 30 Гбит/с в дуплексном режиме;

б) поддержка различных физических линий: медных или оптоволоконных кабелей;

в) связь на базе коммутации пакетов с сохранением целостности данных и управлением потоком;

г) качество обслуживания (*QoS)*;

д) реализованный аппаратно гибкий транспортный механизм;

е) оптимизированный программный интерфейс и удаленный прямой доступ в память (RDMA);

ж) инфраструктура управления, поддерживающая функции отказоустойчивости, аварийного переключения и «горячей» замены.

В спецификации InfiniBand определен чрезвычайно гибкий и масштабированный физический уровень, что обеспечивает возможности последующего наращивания пропускной способности и добавления новых типов поддерживаемых физических линий.

В операционной системе Linux стек протоколов Infiniband реализуется посредством программного пакета *OFED* (OpenFabrics Enterprise Distribution) [6] с сайта www.openfabrics.org (эта организация занимается разработкой и стандартизацией драйверов и пользовательского окружения для Infiniband). На рис. 3 показано местоположение протокола SRP в стеке протоколов Infiniband.

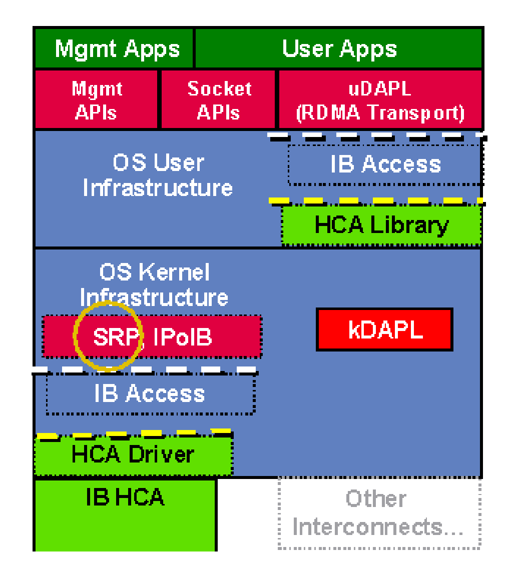


Рисунок 3 – Стек протоколов Infiniband

В SRP протоколе выделяют двух участников. Это серверы – «SRP targets» и клиенты – «SRP initiators». Клиенты «инициаторы» передают SCSI команды и данные через RDMA интерфейс (в нашем случае посредством сети Infiniband) на сервер, а он уже их непосредственно выполняет над своими SCSI-дисками, а результирующие данные отправляет клиентам. Для клиентов это выглядит прозрачно, как если бы они оперировали со своими локальными дисками (рис. 4).

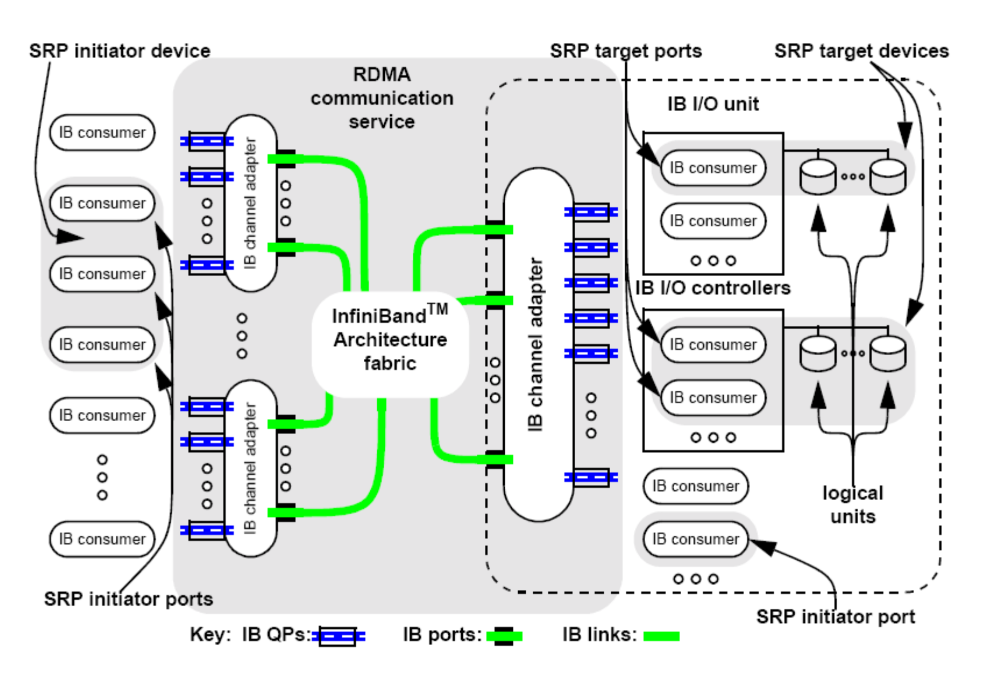


Рисунок 4 – Клиент-серверная модель SRP через Infiniband

На сервере все запросы клиентов обрабатываются посредством SRP target драйвера (*SRPT*). Он работает не напрямую с SCSI драйверами физических устройств, а через специальный SCSI middle level драйвер (*SCST*) [7]. Эта подсистема ядра Linux обеспечивает унифицированный последовательный интерфейс между SCSI target драйверами (в данном случае для драйвера SRPT) и низкоуровневыми SCSI драйверами Linux, что значительно упрощает их разработку (рис. 5).

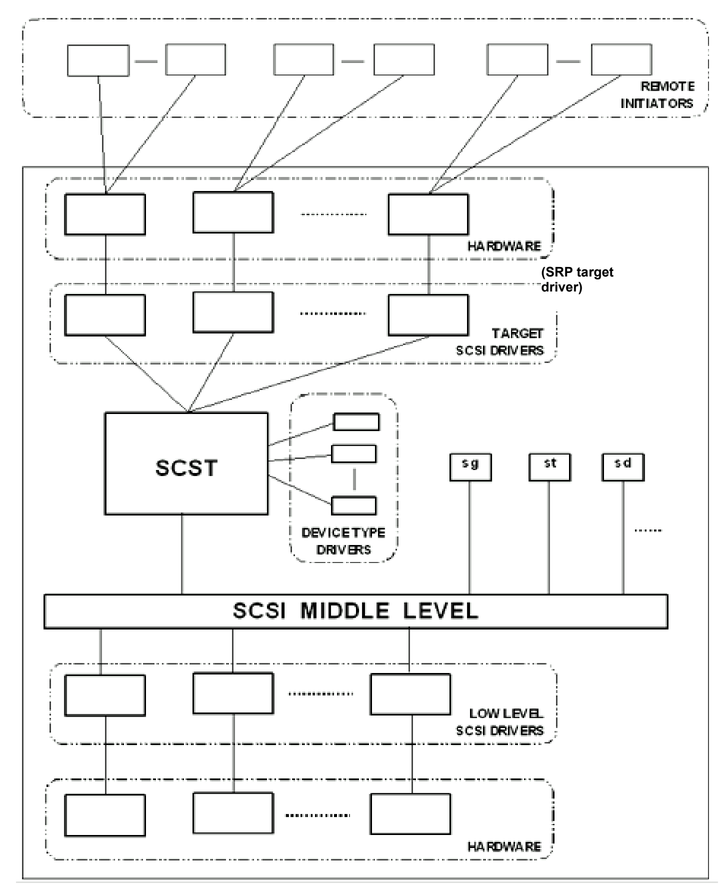


Рисунок 5 – Взаимодействие SCST с SCSI – подсистемой Linux

Кроме того, этот промежуточный драйвер значительно расширяет функциональность сервера, он позволяет экспортировать не только физические SCSI диски, но также программные RAID-массивы и даже обычные файлы-«образы». При этом реальным и виртуальным дискам можно задавать ряд опций экспортирования, а также размер блока данных – *block size* (по умолчанию 512 Б):

1. READ_ONLY – задает доступ только для чтения.
2. WRITE_THROUGH – отключает «*write-back*» кэширование экспортированных дисков. Эта опция снижает производительность, но улучшает надежность хранения данных после случайного обрыва сессии.
3. O_DIRECT – отключает кэширование как записи, так и чтения диска, в текущей версии SCST эта опция в полном объеме не поддерживается.
4. NULLIO – используется для измерений производительности без отсылки команд ввода-вывода к реальным дискам.
5. NV_CACHE – включает долговременное сквозное кэширование данных на виртуальном диске, эта опция улучшает производительность, но значительно снижает надежность хранения данных, поэтому ее целесообразно использовать лишь тогда, когда сервер имеет надежное бесперебойное питание.
6. REMOVABLE – эта опция помечает экспортированный диск как съемный для инициатора.
7. BLOCKIO – задает прямой блочный доступ для экспортированного диска, исключая любое страничное кэширование.

Практическими методами было установлено, что наибольшее быстродействие достигается при использовании блока данных, размером 4096 байт с опцией сквозного кэширования данных NV_CACHE.

Также SCST драйвер позволяет управлять доступом к SCSI дискам на сервере для разных клиентов (*LUN masking*), это дает возможность привязывать к каждому клиенту только его собственный виртуальный диск, без доступа к другим, которые экспортируются сервером.

В общем, главными преимуществами этого протокола являются высокое быстродействие, простота реализации, удобство в настройке, надежность, а недостатком является высокая стоимость. Но поскольку на большинстве суперкомпьютерах кластерного типа, как раз в качестве интерконнекта и используется Infiniband, то это делает технологию SRP чрезвычайно перспективной именно в этой отрасли.

Реализация дисковой подсистемы для кластерного комплекса СКИТ

В 2004 – 2006 гг. в Институте кибернетики НАН Украины был разработан и введенный в эксплуатацию вычислительный комплекс из трех разнородных кластерных суперкомпьютеров для информационных технологий *СКИТ*. Именно на нем и были испытаны преимущества быстродействующей дисковой подсистемы. В инфраструктуре кластерного комплекса создан экспериментальный SRP target сервер на базе 8-ми SATA дисков, объемом 400 Гб каждый, объединенных в *RAID 5* массив, в задачи которого входит экспортирование виртуальных жестких дисков по сети Infiniband для 24-узлового бездискового кластера СКИТ-1. На узлах эти диски используются в качестве источника виртуальной памяти. В качестве временной файловой системы для промежуточных результатов вычислений используется файловая система tmpfs.

Первые тесты показали достаточно высокую эффективность данной технологии. Так, скорость последовательного чтения из виртуального диска составляет приблизительно 300 МБ/с (измеренная утилитой *hdparm*), в сравнении, даже для самых производительных жестких дисков этот показатель не превышает 80 МБ/с:

```
# hdparm –tT /dev/sda
/dev/sda:
 Timing cached reads:   2232 MB in  2.00 seconds = 1114.84 MB/sec
 Timing buffered disk reads:  906 MB in  3.00 seconds = 301.98 MB/sec
```

Рассмотрим производительность дисковой подсистемы, в зависимости от количества одновременно задействованных виртуальных дисков на узлах при максимальной загрузке. Ниже в табл. 1 приведены результаты тестирования производительности дисковой подсистемы посредством утилиты *bonnie++* на файле, размером 4 Гб, для сравнения приведены результаты тестирования файловой системы NFS (в тестировании измеряются: пропускная способность – Кбайт/с, использование процессора, частота поиска).

Как видим, по сравнению с NFS, SRP-диск показывает лучшее быстродействие, при полном использовании SRP сервера лишь одним узлом, а вот при одновременном интенсивном использовании четырьмя и больше клиентами, наблюдается значительное падение скорости блочного чтения/записи. Эта тенденция свойственна всем дисковым системам без параллельной архитектуры, и смягчается только при значительном увеличении дисков в массиве сервера. Но, как было сказано выше, использование виртуальной памяти не предусматривает одновременных массовых операции чтения/записи, поэтому такого быстродействия дискового сервера вполне достаточно для 24-узлового кластера.

Таблица 1

Опера-ция	Последовательное чтение			Последовательная запись		
	Посимво-льно	Поблочно	Перезапись	Посимвольно	Поблочно	Случайный поиск
	Кб/с	Кб/с	Кб/с	Кб/с	Кб/с	с
NFS	26665	27907	3134	29215	84975	460,7
SRP (1 узел)	43001	182930	90063	42353	291033	1025
SRP (2 узла)	42044 x 2	76160 x 2	38258 x2	39693 x 2	139362 x 2	182,2
SRP (4 узла)	26531 x 4	28395 x 4	14951 x4	32294 x 4	55102 x 4	95,9
SRP (6 узлов)	17769 x 6	16964 x 6	9651 x6	21213 x 6	30620 x 6	57,6
SRP (8 узлов)	12084 x 8	12698 x 8	6562 x8	10617 x 8	13099 x 8	40,9

Выводы

Представленная концепция дисковой подсистемы на базе высокопродуктивной сети Infiniband, показала свою эффективность при ее реализации на кластерном комплексе СКИТ. Она дополняет существующую систему хранения данных, и, возможно, станет одной из ключевых технологий для построения высокопроизводительных суперкомпьютеров в будущем.

Литература

1. RFC 3720 – Internet Small Computer Systems Interface (iSCSI).
2. Snyder Peter. A Virtual Memory File System. – Sun Microsystems Inc. – 2550 Garcia Avenue. – Mountain View, CA 94043.
3. ANSI INCITS 365. – 2002. – SCSI RDMA Protocol (SRP), working draft.
4. Режим доступа: http://en.wikipedia.org/wiki/RDMA
5. Режим доступа: http://www.mka.ru/?p=44434
6. Режим доступа: http://www.mellanox.com/products/ofed.php
7. Режим доступа: http://scst.sourceforge.net/scst_pg.html

О.Ю. Бандура, А.Л. Головінський, С.Г. Рябчун

Побудова високопродуктивної дискової підсистеми для суперкомп'ютерів кластерної архітектури

Досліджені загальні концепції високопродуктивної дискової підсистеми як неодмінної складової частини суперкомп'ютерів кластерної архітектури, а також розглянута її побудова в складі кластерного комплексу СКІТ.

A.Yu. Bandura, A.L. Golovinsky, S.G. Ryabchun

Construction of a High-efficiency Disk Subsystem for Supercomputers with Cluster Architecture

In article the general concepts of a high-efficiency disk subsystem as indispensable component of supercomputers with cluster architecture are rewieved, and also considered its construction in structure of cluster complex SKIT.

Статья поступила в редакцию 29.07.2008.